Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Currently Amended) A method of processing an interrupt verification support mechanism in a computer system comprising a processor and an input for external interrupts communicatively coupled to the processor, the method comprising ~~the steps~~:

[[(a)]] processing at least one actual instruction in the processor;

comparing data content of a program counter with data content of an interrupt register to determine if the data content of the program counter matches the data content of the interrupt register; and

[[(b)]] if an external interrupt request is received by the processor or if the data content of the program counter matches the data content of the interrupt register, ~~an interrupt pseudo-instruction is received by the processor, comparing data content of a program counter with data content of an interrupt register and~~ replacing the actual instruction in an instruction fetch stage of the processor with [[the]] an interrupt pseudo-instruction ~~when the data content of the program counter matches the data content of the interrupt register, or when an external interrupt is present~~.

2. (Currently Amended) The method of claim 1, further comprising:

processing at least one actual instruction in the processor in an instruction pipeline wherein instructions are processed concurrently by an instruction fetch stage, an instruction decode stage, an instruction issue stage, an instruction execute stage and a result write-back stage.

3. (Cancelled)

4. (Currently Amended) The method of claim 1, wherein ~~further comprising:~~ ~~creating~~ the interrupt pseudo-instruction is created by a co-processor connected to the processor.

5. (Currently Amended) The method of claim 1, further comprising:

simultaneously processing a ~~number~~ plurality of instructions ~~in the processor~~ in an instruction pipeline of the processor, the instruction pipeline including [[with]] several instruction stages ~~each instruction being in a different instruction stage at a time~~.

6. (Currently Amended) The method of claim 1, further comprising:

storing ~~at least~~ information of ~~a program counter of the instruction which is to be interrupted and~~ a sort of interrupt to use in a ~~set of one or more interrupt registers~~ register of the processor.

7. (Cancelled)

8. (Currently Amended) A processor configured to support [[An]] interrupt verification, the processor comprising: ~~support mechanism device for a computer system comprising a processor and~~

a program counter;

a fetch register configured to store at least one actual instruction;

~~an input for external interrupt requests or interrupt pseudo-instructions communicatively coupled to the processor, wherein the device includes a set of one or more~~

an interrupt register configured to store ~~registers each of which contains information, the information including at least a program counter~~ data content associated with an ~~of the~~ instruction ~~which~~ that is to be interrupted; ~~and a sort of interrupt to use, so as to enable the device to process at least one actual instruction,~~

a comparator configured to compare the data content of the interrupt register with data content of the program counter to determine if the data content of the interrupt register matches the data content of the program counter; and

a multiplexer configured to replace the at least one actual instruction with an interrupt pseudo-instruction if an external interrupt request is received by the processor or if the data content of the program counter matches the data content of the interrupt register~~, the at least one actual instruction is replaced with the pseudo-instruction~~.

9. (Currently Amended) The ~~device~~ processor of claim 8 ~~wherein the device~~ further ~~comprises~~ comprising:

an instruction fetch that includes the fetch register, the ~~with a~~ program counter, and [[an]] the interrupt register, the ~~instruction~~ fetch register being coupled to a first input of [[a]] the multiplexer ~~for transmitting instructions to said multiplexer, a second input of the multiplexer connected to an interrupt pseudo-instruction input and the program counter connected with the interrupt register by a comparator~~.

10. (Currently Amended) The ~~device~~ processor of claim 9, wherein [[the]] a second input of the multiplexer is ~~capable of receiving~~ coupled to the interrupt pseudo-instruction ~~signals or external interrupt requests~~.

11. (Currently Amended) The ~~device~~ processor of claim 9, wherein the comparator creates a high level signal [[only]] if the data content of the program counter matches the data content of the interrupt register.

12. (Currently Amended) The ~~device~~ processor of claim 9, wherein an output of the comparator is connected to a first input of an or-operator, and a second input of the or-operator is connected to an interrupt controller ~~so as~~ to enable the or-operator to create a high level signal if a signal received from the interrupt controller differs from a signal received from the comparator.

13-14. (Cancelled)

15. (Currently Amended) The ~~device~~ processor of claim 9, wherein an instruction coming from an output of the multiplexer is sequentially processed in an instruction pipeline of the processor.

16. (Currently Amended) The ~~device~~ processor of claim 9, wherein an instruction pipeline of the processor includes an instruction fetch stage, an instruction decode stage, an instruction issue stage, an instruction execute stage and a result write-back stage.

17. (Currently Amended) The ~~device~~ processor of claim 9, wherein the interrupt pseudo-instruction effects instruction state stages required by the interrupt pseudo-instruction.

18. (Currently Amended) The ~~device~~ processor of claim 16, wherein if an interrupt request or an interrupt pseudo-instruction is received by the processor, the processor is ~~adapted~~ configured to cancel an instruction that is in the instruction fetch stage ~~when the interrupt request or the interrupt pseudo-instruction is received~~ and to reissue the cancelled instruction starting at the instruction fetch stage.

19. (Currently Amended) The ~~device~~ processor of claim 16, wherein if an interrupt request or an interrupt pseudo-instruction is received by the processor, the processor is ~~adapted~~ configured to cancel an instruction that is in any instruction stage ~~when the interrupt request or the interrupt pseudo-instruction is received~~ and to reissue the cancelled instruction starting at the instruction fetch stage.

20. (Currently Amended) The ~~device~~ processor of claim 8, wherein the interrupt pseudo-instruction is created by a co-processor connected to the processor.

21. (Currently Amended) The ~~device~~ processor of claim 20, wherein ~~the device is a media decoding system,~~ the processor is a core decoder processor and the co-processor is a decoding accelerator ~~adapted~~ configured to assist the core processor with a decoding function.

22. (Currently Amended) The ~~device~~ processor of claim 20, wherein the processor is a reduced instruction set computer (RISC) processor.

23-24. (Cancelled)